ManifoldNeRF: View-dependent Image Feature Supervision for Few-shot Neural Radiance Fields

Daiju Kanaoka[1,2]
kanaoka.daiju327@mail.kyutech.jp
Motoharu Sonogashira[2]
motoharu.sonogashira@riken.jp
Hakaru Tamukoh[1,3]
tamukoh@brain.kyutech.ac.jp
Yasutomo Kawanishi[2]
yasutomo.kawanishi@riken.jp

[1] Kyushu Institute of Technology
Fukuoka, Japan
[2] RIKEN Guardian Robot Project
Kyoto, Japan
[3] Research Center for Neuromorphic AI Hardware
Fukuoka, Japan

1 Experimental details

The proposed method was trained by following a training process shown in Algorithm 1. The hyperparameters in the training process, manifold loss interval K and scaling factor λ, were set to 10 and 0.1, respectively. These are the same values as used in the official implementation of DietNeRF [1].

The IDs of the known viewpoints used in the randomly selected experiments from the NeRF synthetic dataset [4] were $[2, 16, 26, 55, 73, 75, 86, 93]$, and those used in the DTU MVS dataset were $[0, 6, 7, 23, 32, 37, 39, 48]$.

2 Analysis of feature vector obtained from pre-trained feature extractor

In this section, we describe experiments conducted to verify the changes in feature vectors with a change in viewpoints. For this experiment, we generated 36 images rendered by rotating the camera position by 10 degrees around an axis of the LEGO scene in the NeRF synthetic dataset. We input these images to the vision encoder of the CLIP to obtain the feature vectors. The obtained feature vectors were projected onto a 2D space using UMAP and visualized in the 2D space to confirm the changes in feature vectors along with continuous changes in viewpoints.

The experimental results are shown in Fig. 1. The feature vectors of adjacent viewpoints are located in the neighborhood, indicating that the feature vectors change continuously as the viewpoints change, as claimed by the Parametric Eigenspace [5].

Algorithm 1: Training process of ManifoldNeRF

Data: Known viewpoints $\mathcal{D} = \{(I, \mathbf{p})\}$, a pre-trained feature extractor $\phi(\cdot)$,
threshold of distance between viewpoints ε, manifold loss interval K, scaling factor λ, batch size $|\mathcal{R}|$,
learning rate η_i, MSE loss $\mathcal{L}_{\mathrm{MSE}}$, manifold loss $\mathcal{L}_{\mathrm{ML}}$
Result: Trained Neural Radiance Field $f_\theta(\cdot,\cdot)$

Initialize NeRF $f_\theta(\cdot,\cdot)$;
Pre-compute feature vectors $\mathcal{V} = \{\mathbf{v} = \phi(I) : (I, \mathbf{p}) \in \mathcal{D}\}$;
Pre-compute pairs of viewpoint
$\mathcal{P} = \{(\{\mathbf{p}_{k,1}, \mathbf{v}_{k,1}\}, \{\mathbf{p}_{k,2}, \mathbf{v}_{k,2}\}) : (I_{k,1}, \mathbf{p}_{k,1}), (I_{k,2}, \mathbf{p}_{k,2}) \in \mathcal{D}, \mathbf{v}_{k,1}, \mathbf{v}_{k,2} \in \mathcal{V}\ , if\ |\mathbf{p}_{k,1} - \mathbf{p}_{k,2}| < \varepsilon\}$;
for *i from 1 to num_iters* **do**
    Sample ray batch $\mathcal{R}$, ground-truth colors $\mathbf{C}(\cdot)$;
    Render rays $\widehat{\mathbf{C}}(\cdot)$;
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{MSE}}(\mathcal{R}, \mathbf{C}, \widehat{\mathbf{C}})$;
    if $i \ \%\ K = 0$ **then**
        Sample pair of viewpoints $(\{\mathbf{p}_k^1, \mathbf{v}_{k,1}\}, \{\mathbf{p}_{k,2}, \mathbf{v}_{k,2}\}) \sim \mathcal{P}$;
        Compute interpolation coefficient s;
        Compute unknown viewpoint $\widehat{\mathbf{p}}_u = SLERP(\mathbf{p}_{k,1}, \mathbf{p}_{k,2}, s)$;
        Render image $\widehat{I}$ at viewpoint $\widehat{\mathbf{p}}_u$;
        Compute feature vector of $\widehat{I}$: $\widehat{\mathbf{v}}_u = \phi(\widehat{I})$;
        Interpolate feature vector $\mathbf{v}_u = LERP(\mathbf{v}_{k,1}, \mathbf{v}_{k,2}, s)$
        $\mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_{\mathrm{ML}}(\mathbf{v}_u, \widehat{\mathbf{v}}_u)$;
    end
    Update parameters: $\theta \leftarrow Adam(\theta, \eta_i, \nabla_\theta \mathcal{L})$;
end

Table 1: Results of performance change with different number of training data. The training dataset is a LEGO scene from the NeRF synthetic dataset.

NeRF	PSNR ↑	SSIM ↑	LPIPS ↓
4	9.035	0.504	0.468
8	9.727	0.521	0.469
12	10.001	0.552	0.455
16	27.613	0.926	0.065

ManifoldNeRF	PSNR ↑	SSIM ↑	LPIPS ↓
4	9.219	0.526	0.464
8	21.607	0.806	0.174
12	25.215	0.874	0.108
16	25.793	0.884	0.102

3 Performance with different numbers of training data

In the experiments conducted in Sec. 3 of the paper, the feature vectors between viewpoints differing by 90 degrees were calculated to be close to the ground truth. Therefore, we assumed that the performance of the proposed method would be higher when we selected 8 images if we prepared viewpoints that differ by 90 degrees from each other in the horizontal and the diagonal viewpoints.

In this section, we evaluated the change in performance when the number of training data is changed. Table 1 shows the experimental results when we changed the training data for NeRF and ManifoldNeRF to 4, 8, 12, and 16. NeRF perform poorly when the training data is less than 16. However, the performance of ManifoldNeRF increased significantly when the training data was 8 and above.

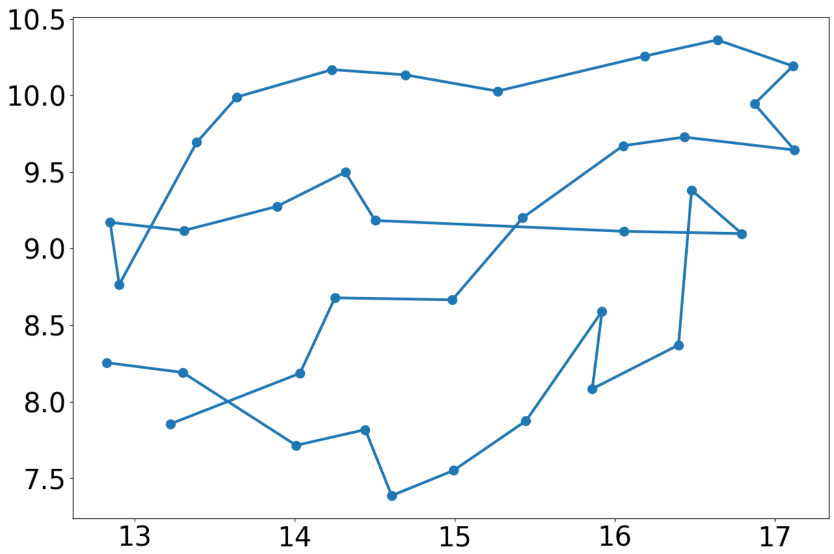


Figure 1: Projected feature vectors extracted with the pre-trained feature extractor into a two-dimensional space using UMAP. The dots in the graph denote the projected feature vectors, and the lines connect the dots corresponding to adjacent viewpoints.

4 Details of experimental results using the DTU MVS dataset

Table 2 and Fig. 2 show the experimental results for 8 scenes selected from the DTU MVS dataset [2]. ManifoldNeRF performed well in 4 scenes of the 8 scenes. However, in the remaining 4 scenes, the performance was comparable to that of vanilla NeRF. The reason why the proposed method sometimes performed not well is that the performance of Maniofld-NeRF is strongly dependent on the location of known viewpoints. In contrast, InfoNeRF [3] performed more stably than the other methods.

Next, Table 3 and Fig. 3 show the results of fine-tuning the model trained with InfoNeRF using ManifoldNeRF. We confirmed the performance improvement by fine-tuning the model trained with InfoNeRF. The reason for the performance improvement is that InfoNeRF applies constraints to each point on the ray, whereas ManifoldNeRF applies constraints to the feature vectors of the image obtained from the viewpoints between neighbouring viewpoints, and the optimization targets are different. From the above, we demonstrate that even when the known viewpoints are random, combining other methods with the proposed method can improve performance.

Table 2: Results of training 8 randomly selected images in 8 scene of the DTU MVS dataset. The highest score is in bold, and the second-highest score is underlined.

#6	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	15.550	0.460	0.472
InfoNeRF	13.352	0.397	0.462
DietNeRF	15.210	0.426	0.476
ManifoldNeRF (ours)	**16.232**	**0.508**	**0.451**

#65	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	11.970	0.481	0.527
InfoNeRF	14.786	0.484	0.431
DietNeRF	20.883	0.698	0.352
ManifoldNeRF (ours)	**22.197**	**0.702**	**0.302**

#30	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	8.054	0.491	0.560
InfoNeRF	**17.657**	**0.663**	**0.254**
DietNeRF	6.092	0.298	0.675
ManifoldNeRF (ours)	6.406	0.387	0.633

#45	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	7.558	0.220	0.710
InfoNeRF	**10.719**	**0.422**	**0.441**
DietNeRF	7.097	0.216	0.662
ManifoldNeRF (ours)	7.418	0.181	0.721

#56	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	21.484	0.621	**0.353**
InfoNeRF	18.644	0.477	0.474
DietNeRF	19.026	0.538	0.427
ManifoldNeRF (ours)	**22.197**	**0.639**	0.367

#114	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	18.691	0.636	0.396
InfoNeRF	21.382	0.611	0.364
DietNeRF	20.861	0.673	0.337
ManifoldNeRF (ours)	**23.202**	**0.732**	**0.299**

#41	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	8.236	0.312	0.636
InfoNeRF	**14.681**	**0.484**	**0.423**
DietNeRF	8.36	0.246	0.636
ManifoldNeRF (ours)	8.963	0.317	0.628

#61	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	7.793	0.236	0.684
InfoNeRF	**14.634**	**0.543**	**0.395**
DietNeRF	11.974	0.463	0.508
ManifoldNeRF (ours)	7.518	0.267	0.679

References

[1] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, October 2021.

[2] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 406–413, June 2014.

[3] Mijeong Kim, Seonguk Seo, and Bohyung Han. InfoNeRF: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921, June 2022.

[4] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Computer Vision – ECCV 2020*, pages 405–421, 2020.

[5] Hiroshi Murase and Shree K Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

Table 3: Results of fine-tuning the model trained with InfoNeRF using ManifoldNeRF. The results w/o fine-tuning are the same as for InfoNeRF in Table 2. The highest score is in bold, and the second-highest score is underlined.

#6	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	13.352	0.397	0.462
20k iters	15.157	0.438	0.491
40k iters	15.210	0.448	0.476
60k iters	15.087	0.461	0.458s
80k iters	15.211	0.471	0.458
100k iters	**15.398**	**0.472**	**0.448**

#30	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	17.657	0.663	0.254
20k iters	20.335	0.808	0.197
40k iters	**20.412**	0.828	0.185
60k iters	20.364	**0.839**	0.183
80k iters	20.359	0.835	0.180
100k iters	20.285	0.837	**0.179**

#41	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	14.681	0.484	0.423
20k iters	17.032	0.570	0.443
40k iters	**17.111**	0.585	0.427
60k iters	16.890	0.582	0.422
80k iters	16.904	0.595	0.411
100k iters	17.031	**0.599**	**0.405**

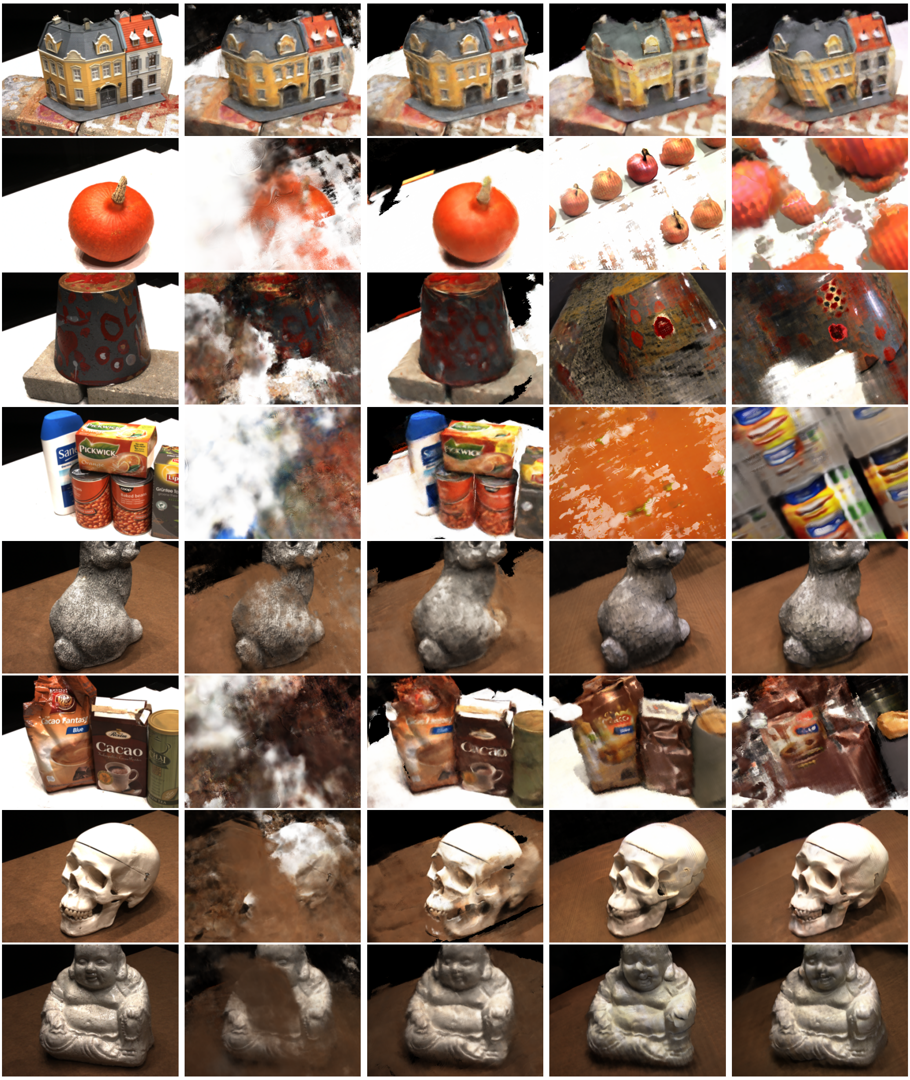
#45	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	10.719	0.422	0.441
20k iters	14.867	0.504	0.412
40k iters	14.882	0.514	0.399
60k iters	14.842	0.520	0.390
80k iters	14.898	**0.523**	0.389
100k iters	**14.903**	**0.523**	**0.383**

#56	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	18.644	0.477	0.474
20k iters	19.912	0.510	0.472
40k iters	**20.130**	0.528	0.462
60k iters	19.880	0.532	0.458
80k iters	19.773	0.531	0.453
100k iters	19.902	**0.540**	**0.451**

#61	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	14.634	0.543	0.395
20k iters	15.588	0.544	0.437
40k iters	**15.926**	0.554	0.420
60k iters	15.772	0.563	0.415
80k iters	15.861	**0.568**	0.409
100k iters	15.799	0.560	**0.403**

#65	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	14.786	0.484	0.431
20k iters	20.468	0.658	0.369
40k iters	20.615	0.673	0.349
60k iters	20.654	0.687	0.339
80k iters	20.768	0.693	0.334
100k iters	**20.821**	**0.705**	**0.331**

#114	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
w/o fine-tuning	21.382	0.611	0.364
20k iters	21.902	0.658	0.372
40k iters	**22.092**	0.672	0.358
60k iters	21.948	**0.677**	0.350
80k iters	21.791	0.673	0.350
100k iters	21.696	0.675	**0.346**



Ground Truth | NeRF | InfoNeRF | DietNeRF | ManifoldNeRF (ours)

Figure 2: Qualitative comparison on 8 scenes of the MVS DTU dataset

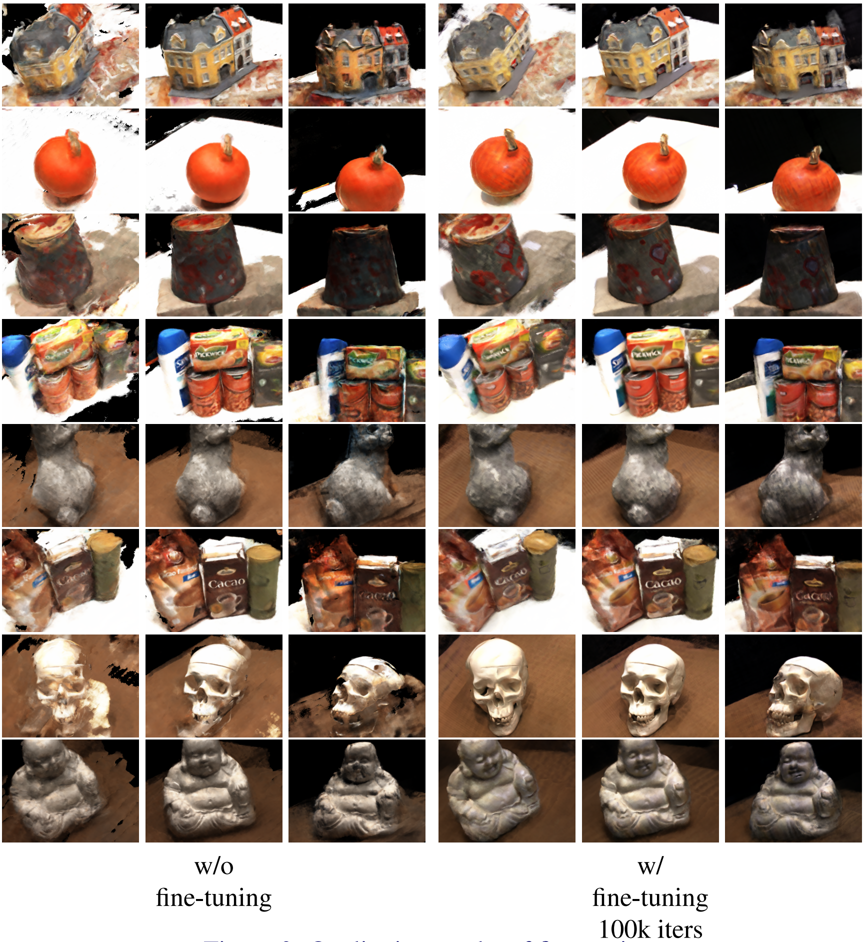


Figure 3: Qualitative results of fine-tuning.